

Simulation of flows with violent free surface motion and moving objects using unstructured grids

Rainald Löhner^{1,*}, Chi Yang^{1,2,†} and Eugenio Oñate^{3,§}

¹*School of Computational Science and Informatics, M.S. 4C7, George Mason University, Fairfax, VA, U.S.A.*

²*Chang Jiang Scholar, Shanghai Jiao Tong University, China*

³*International Center for Numerical Methods in Engineering (CIMNE), Universidad Politécnica de Cataluña, Barcelona, Spain*

SUMMARY

A volume of fluid (VOF) technique has been developed and coupled with an incompressible Euler/Navier–Stokes solver operating on adaptive, unstructured grids to simulate the interactions of extreme waves and three-dimensional structures. The present implementation follows the classic VOF implementation for the liquid–gas system, considering only the liquid phase. Extrapolation algorithms are used to obtain velocities and pressure in the gas region near the free surface. The VOF technique is validated against the classic dam-break problem, as well as series of 2D sloshing experiments and results from SPH calculations. These and a series of other examples demonstrate that the ability of the present approach to simulate violent free surface flows with strong nonlinear behaviour. Copyright © 2006 John Wiley & Sons, Ltd.

Received 3 October 2005; Revised 9 March 2006; Accepted 19 March 2006

KEY WORDS: marine engineering; computational techniques; incompressible flow; projection schemes; VOF; level set; FEM; CFD

1. INTRODUCTION

High sea states, waves breaking near shores and moving ships, the interaction of extreme waves with floating structures, green water on deck and sloshing (e.g. in liquid natural gas (LNG) tankers) are but a few examples of flows with violent free surface motion. Many of these flows have a profound impact on marine engineering.

*Correspondence to: Rainald Löhner, School of Computational Science and Informatics, M.S. 4C7, George Mason University, Fairfax, VA, U.S.A.

†E-mail: rlohner@science.gmu.edu

‡E-mail: cyang@gmu.edu

§E-mail: onate@cimne.upc.edu

The computation of highly nonlinear free surface flows is difficult because neither the shape nor the position of the interface between air and water is known *a priori*; on the contrary, it often involves unsteady fragmentation and merging processes. There are basically two approaches to compute flows with free surface: interface-tracking and interface-capturing methods. The former computes the liquid flow only, using a numerical grid that adapts itself to the shape and position of the free surface. The free surface is represented and tracked explicitly either by marking it with special marker points, or by attaching it to a mesh surface. Various surface fitting methods for attaching the interface to a mesh surface were developed during the past decades using the finite element method. In the interface-tracking methods, the free surface is treated as a boundary of the computational domain, where the kinematic and dynamic boundary conditions are applied. These methods cannot be used if the interface topology changes significantly, as is contemplated here for overturning or breaking waves. The second possible approach is given by the so-called interface-capturing methods [1–13]. These consider both fluids as a single effective fluid with variable properties; the interface is captured as a region of sudden change in fluid properties. The main problem of complex free surface flows is that the density ρ jumps by three orders of magnitude between the gaseous and liquid phase. Moreover, this surface can move, bend and reconnect in arbitrary ways. In order to illustrate the difficulties that can arise if one treats the complete system, consider a hydrostatic flow, where the exact solution is $\mathbf{v} = 0$, $p = -\rho \mathbf{g} \cdot (\mathbf{x} - \mathbf{x}_0)$, where \mathbf{x}_0 denotes the position of the free surface. Unless the free surface coincides with the faces of elements, there is no way for typical finite element shape functions to capture the discontinuity in the gradient of the pressure. This implies that one has to either increase the number of Gauss-points [14] or modify (e.g. enrich) the shape-function space [13]. Using the standard linear element procedure leads to spurious velocity jumps at the interface, as any small pressure gradient that ‘pollutes over’ from the water to the air region will accelerate the air considerably. This in turn will lead to loss of divergence, causing more spurious pressures. The whole cycle may, in fact, lead to a complete divergence of the solution. Faced with this dilemma, most flows with free surfaces have been solved neglecting the air. This approach neglects the pressure build-up due to volumes of gas enclosed by liquid, and therefore is not universal. However, in the present case, we have followed this approach, fully aware of the limitations.

The remainder of the paper is organized as follows: Section 2 summarizes the basic elements of the present incompressible flow solver; Sections 3 and 4 describe the temporal and spatial discretization; Section 5 describes the volume of fluid extensions; some examples are shown in Section 6; finally, some conclusions are given in Section 7.

2. BASIC ELEMENTS OF THE SOLVER

In order to fix the notation, the equations describing incompressible, Newtonian flows in an arbitrary Lagrangian–Eulerian (ALE) frame are written as

$$\rho \mathbf{v}_{,t} + \rho \mathbf{v}_a \nabla \mathbf{v} + \nabla p = \nabla \mu \nabla \mathbf{v} + \rho \mathbf{g} \quad (1)$$

$$\nabla \cdot \mathbf{v} = 0 \quad (2)$$

Here ρ denotes the density, \mathbf{v} the velocity vector, p the pressure, μ the viscosity and \mathbf{g} the gravity vector. The advective velocity is given by $\mathbf{v}_a = \mathbf{v} - \mathbf{w}$, where \mathbf{w} is the mesh velocity. We remark that both the gaseous and liquid phases are considered incompressible, thus Equation (2). However,

we will consider compressibility for bubbles, but will treat the compressibility in a global manner. The liquid–gas interface is described by a scalar equation of the form

$$\Phi_{,t} + \mathbf{v}_a \cdot \nabla \Phi = 0 \quad (3)$$

For the classic volume of fluid (VOF) technique, Φ represents the percentage of liquid in a cell/element or control volume (see References [1, 2, 7–10, 12]). For pseudo-concentration (PC) techniques, Φ represents the total density of the material in a cell/element or control volume. For the level set (LS) approach Φ represents the signed distance to the interface [11].

Since over a decade [15–18] the numerical schemes chosen to solve the incompressible Navier–Stokes equations given by Equations (1) and (2) have been based on the following criteria:

- spatial discretization using *unstructured grids* (in order to allow for arbitrary geometries and adaptive refinement);
- spatial approximation of unknowns with *simple finite elements* (in order to have a simple input/output and code structure);
- temporal approximation using *implicit integration of viscous terms and pressure* (the interesting scales are the ones associated with advection);
- temporal approximation using *explicit integration of advective terms*;
- *low-storage, iterative solvers* for the resulting systems of equations (in order to solve large 3D problems); and
- steady results that are *independent from the timestep* chosen (in order to have confidence in convergence studies).

3. TEMPORAL DISCRETIZATION

For most of the applications listed above, the important physical phenomena propagate with the *advective* timescales. We will therefore assume that the advective terms require an explicit time integration. Diffusive phenomena typically occur at a much faster rate, and can/should therefore be integrated implicitly. Given that the pressure establishes itself immediately through the pressure–Poisson equation, an implicit integration of pressure is also required. The hyperbolic character of the advection operator and the elliptic character of the pressure–Poisson equation have led to a number of so-called projection schemes. The key idea is to predict first a velocity field from the current flow variables without taking the divergence constraint into account. In a second step, the divergence constraint is enforced by solving a pressure–Poisson equation. The velocity increment can therefore be separated into an advective–diffusive and pressure increment:

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta \mathbf{v}^a + \Delta \mathbf{v}^p = \mathbf{v}^* + \Delta \mathbf{v}^p \quad (4)$$

For an explicit (forward Euler) integration of the advective terms, with implicit integration of the viscous terms, one complete timestep is given by:

- Advective–diffusive prediction: $\mathbf{v}^n \rightarrow \mathbf{v}^*$

$$\left[\frac{\rho}{\Delta t} - \theta \nabla \mu \nabla \right] (\mathbf{v}^* - \mathbf{v}^n) + \mathbf{v}_a^n \cdot \nabla \mathbf{v}^n + \nabla p^n = \nabla \mu \nabla \mathbf{v}^n + \rho \mathbf{g} \quad (5)$$

- Pressure correction: $p^n \rightarrow p^{n+1}$

$$\nabla \cdot \mathbf{v}^{n+1} = 0 \quad (6)$$

$$\rho \frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} + \nabla(p^{n+1} - p^n) = 0 \quad (7)$$

which results in

$$\nabla \cdot \frac{1}{\rho} \nabla(p^{n+1} - p^n) = \frac{\nabla \cdot \mathbf{v}^*}{\Delta t} \quad (8)$$

- Velocity correction: $\mathbf{v}^* \rightarrow \mathbf{v}^{n+1}$

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \frac{\Delta t}{\rho} \nabla(p^{n+1} - p^n) \quad (9)$$

At steady state, $\mathbf{v}^* = \mathbf{v}^n = \mathbf{v}^{n+1}$ and the residuals of the pressure correction vanish, implying that the result does not depend on the timestep Δt . θ denotes the implicitness factor for the viscous terms ($\theta = 1$: first-order, fully implicit, $\theta = 0.5$: second-order, Crank–Nicholson). One can replace the one-step explicit advective–diffusive predictor by a multistage Runge–Kutta scheme [19], allowing for higher accuracy in the advection-dominated regions and larger timesteps without a noticeable increment in CPU cost.

A k -step, time-accurate Runge–Kutta scheme of order k for the advective parts may be written as

$$\rho \mathbf{v}^i = \rho \mathbf{v}^n + \alpha^i \gamma \Delta t \left(-\rho \mathbf{v}_a^{i-1} \cdot \nabla \mathbf{v}^{i-1} - \nabla p^n + \nabla \mu \nabla \mathbf{v}^{i-1} \right); \quad i = 1, k-1 \quad (10)$$

$$\left[\frac{\rho}{\Delta t} - \theta \nabla \mu \nabla \right] (\mathbf{v}^k - \mathbf{v}^n) + \rho \mathbf{v}_a^{k-1} \cdot \nabla \mathbf{v}^{k-1} + \nabla p^n = \nabla \mu \nabla \mathbf{v}^{k-1} \quad (11)$$

Here, the α^i are the standard Runge–Kutta coefficients $\alpha^i = 1/(k+1-i)$. As compared to the original scheme given by Equation (5), the $k-1$ stages of Equation (10) may be seen as a predictor (or replacement) of \mathbf{v}^n by \mathbf{v}^{k-1} . The original right-hand side has not been modified, so that at steady state $\mathbf{v}^n = \mathbf{v}^{k-1}$, preserving the requirement that the steady state be independent of the timestep Δt . The factor γ denotes the local ratio of the stability limit for explicit timestepping for the viscous terms versus the timestep chosen. Given that the advective and viscous timestep limits are proportional to

$$\Delta t_a \approx \frac{h}{|\mathbf{v}|}; \quad \Delta t_v \approx \frac{\rho h^2}{\mu} \quad (12)$$

we immediately obtain

$$\gamma = \frac{\Delta t_v}{\Delta t_a} \approx \frac{\rho |\mathbf{v}| h}{\mu} \approx Re_h \quad (13)$$

or, in its final form:

$$\gamma = \min(1, Re_h) \quad (14)$$

In regions away from boundary layers, this factor is $O(1)$, implying that a high-order Runge–Kutta scheme is recovered. Conversely, for regions where $Re_h = O(0)$, the scheme reverts back to the original one (Equation (5)). Projection schemes of this kind (explicit advection with

a variety of schemes, implicit diffusion, pressure-Poisson equation for either the pressure or pressure increments) have been widely used in conjunction with spatial discretizations based on finite differences [20–23], finite volumes [24], and finite elements [15–19, 25–34].

One *complete timestep* is then comprised of the following substeps:

- predict velocity (advective–diffusive predictor, Equations (5), (10) and (11));
- extrapolate the pressure (imposition of boundary conditions);
- update the pressure (Equation (8));
- correct the velocity field (Equation (9));
- extrapolate the velocity field; and
- update the scalar interface indicator.

4. SPATIAL DISCRETIZATION

As stated before, we desire a spatial discretization with unstructured grids in order to:

- Approximate arbitrary domains, and
- Perform adaptive refinement in a straightforward manner, i.e. without changes to the solver.

From a numerical point of view, the difficulties in solving Equations (1)–(3) are the usual ones. First-order derivatives are problematic (overshoots, oscillations, instabilities), while second-order derivatives can be discretized by a straightforward Galerkin approximation. We will first treat the advection operator and then proceed to the divergence operator. Given that tetrahedral grids solvers based on edge data structures incur a much lower indirect addressing and CPU overhead than those based on element data structures [35], only these will be considered.

4.1. The advection operator

It is well known that a straightforward Galerkin approximation of the advection terms will lead to an unstable scheme (recall that on a 1D mesh of elements with constant size, the Galerkin approximation is simply a central difference scheme). Three ways have emerged to modify (or stabilize) the Galerkin discretization of the advection terms:

- integration along characteristics [36, 37];
- Taylor–Galerkin (or streamline diffusion) [26, 38, 39], and
- edge-based upwinding [18].

Of these, we only consider the third option here. The Galerkin approximation for the advection terms yields a right-hand side of the form

$$r^i = D^{ij} \mathcal{F}_{ij} = D^{ij} (\mathbf{f}_i + \mathbf{f}_j) \quad (15)$$

where the \mathbf{f}_i are the ‘fluxes along edges’

$$\mathbf{f}_i = S_k^{ij} \mathbf{F}_i^k, \quad S_k^{ij} = \frac{d_k^{ij}}{D^{ij}}, \quad D^{ij} = \sqrt{d_k^{ij} d_k^{ij}} \quad (16)$$

$$\mathcal{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j, \quad \mathbf{f}_i = (S_k^{ij} v_i^k) \mathbf{v}_i, \quad \mathbf{f}_j = (S_k^{ij} v_j^k) \mathbf{v}_j \quad (17)$$

and the edge-coefficients are based on the shape-functions N^i as follows:

$$d_k^{ij} = \frac{1}{2} \int_{\Omega} (N_{,k}^i N^j - N_{,k}^j N^i) d\Omega \quad (18)$$

A consistent numerical flux is given by

$$\mathcal{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |v^{ij}|(\mathbf{v}_i - \mathbf{v}_j), \quad v^{ij} = \frac{1}{2} S_k^{ij} (v_i^k + v_j^k) \quad (19)$$

As with all other edge-based upwind fluxes, this first-order scheme can be improved by reducing the difference $\mathbf{v}_i - \mathbf{v}_j$ through (limited) extrapolation to the edge centre [35]. The same scheme is used for the transport equation that describes the propagation of the VOF fraction, PC or distance to the free surface given by Equation (3).

4.2. The divergence operator

A persistent difficulty with incompressible flow solvers has been the derivation of a stable scheme for the divergence constraint (2). The stability criterion for the divergence constraint is also known as the Ladyzenskaya–Babuska–Brezzi or LBB condition [40]. The classic way to satisfy the LBB condition has been to use different functional spaces for the velocity and pressure discretization [41]. Typically, the velocity space has to be richer, containing more degrees of freedom than the pressure space. Elements belonging to this class are the p1/p1+bubble mini-element [42], the p1/iso-p1 element [43], and the p1/p2 element [44]. An alternative way to satisfy the LBB condition is through the use of artificial viscosities [15], ‘stabilization’ [45–47] or a ‘consistent numerical flux’ (more elegant terms for the same thing). The equivalency of these approaches has been repeatedly demonstrated (e.g. References [15, 35, 42]). The approach taken here is based on consistent numerical fluxes, as it fits naturally into the edge-based framework. For the divergence constraint, the Galerkin approximation along edge i, j is given by

$$\mathcal{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j, \quad \mathbf{f}_i = S_k^{ij} v_i^k, \quad \mathbf{f}_j = S_k^{ij} v_j^k \quad (20)$$

A consistent numerical flux may be constructed by adding pressure terms of the form:

$$\mathcal{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |\lambda^{ij}|(p_i - p_j) \quad (21)$$

where the eigenvalue λ^{ij} is given by the ratio of the characteristic advective timestep of the edge Δt and the characteristic advective length of the edge l :

$$\lambda^{ij} = \frac{\Delta t^{ij}}{l^{ij}} \quad (22)$$

Higher-order schemes can be derived by reconstruction and limiting, or by substituting the first-order differences of the pressure with third-order differences:

$$\mathcal{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |\lambda^{ij}| \left(p_i - p_j + \frac{l^{ij}}{2} (\nabla p_i + \nabla p_j) \right) \quad (23)$$

This results in a stable, low-diffusion, fourth-order damping for the divergence constraint.

5. VOLUME OF FLUID EXTENSIONS

The extension of a solver for the incompressible Navier–Stokes equations to handle free surface flows via the VOF or LS techniques requires a series of extensions which are the subject of the present section. Before going on, we remark that both the VOF and LS approaches were implemented as part of this effort. Experience indicates that both work well. For VOF, it is important to have a monotonicity preserving scheme for Ψ . For LS, it is important to balance the cost and accuracy loss of reinitializations *vis a vis* propagation. Given that the advection solvers used are all monotonicity preserving, and that the VOF option is less CPU-demanding than LS, only the VOF technique is considered in the following. In what follows, we will assume that Φ is bounded by values for liquid and gas (e.g. $0 \leq \Phi \leq 1$ for VOF, $\rho_g \leq \Phi \leq \rho_l$ for PC) and that the liquid–gas interface is defined by the average of these extreme values (i.e. $\Phi = 0.5$ for VOF, $\Phi = 0.5 \cdot (\rho_g + \rho_l)$ for PC, $\Phi = 0$ for LS).

5.1. Extrapolation of the pressure

The pressure in the gas region needs to be extrapolated in order to obtain the proper velocities in the region of the free surface. This extrapolation is performed using a three-step procedure. In the first step, the pressures for all points in the gas region are set to (constant) values, either the atmospheric pressure or, in the case of bubbles, the pressure of the particular bubble. In a second step, the gradient of the pressure for the points in the liquid that are close to the liquid–gas interface are extrapolated from the points inside the liquid region (see Figure 1). This step is required as the pressure gradient for these points cannot be computed properly from the data given. Using this information (i.e. pressure and gradient of pressure), the pressure for the points in the gas that are close to the liquid–gas interface are computed.

5.2. Extrapolation of the velocity

The velocity in the gas region needs to be extrapolated properly in order to propagate accurately the free surface. This extrapolation is started by initializing all velocities in the gas region to $\mathbf{v} = 0$. Then, for each subsequent layer of points in the gas region where velocities have not been

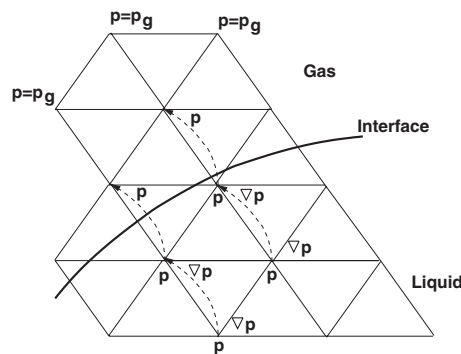


Figure 1. Extrapolation of the pressure.

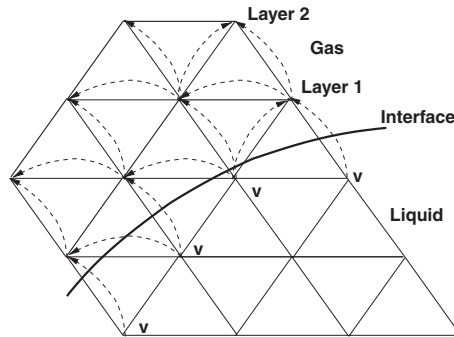


Figure 2. Extrapolation of the velocity.

extrapolated (unknown values), an average of the velocities of the surrounding points with known values is taken (see Figure 2).

5.3. Keeping interfaces sharp

The VOF and PC options propagate Heavyside functions through an Eulerian mesh. The ‘sharpness’ of such profiles requires the use of monotonicity preserving schemes for advection, such as total variation diminishing (TVD) or flux-corrected transport (FCT) techniques [35]. Level set methods propagate a linear function, numerically a much simpler problem. Regardless of the technique used, one finds that shear and vortical flowfields will tend to smooth and distort Φ . Fortunately, both TVD and FCT algorithms allow for limiters that keep the solution monotonic while enhancing the sharpness of the solution. For the TVD schemes Roe’s Super-B limiter [48] produces the desired effect. For FCT one increases the anti-diffusion by a small fraction (e.g. $c = 1.01$). The limiting procedure keeps the solution monotonic, while the increased anti-diffusion steepens Φ as much as possible on a mesh. With these schemes, the discontinuity in Φ is captured within 1–2 gridpoints for all times. For LS the distance-function Φ must be reinitialized periodically so that it truly represents the distance to the liquid–gas interface.

5.4. Imposition of constant mass

Experience indicates that the amount of liquid mass (as measured by the region where the VOF indicator is larger than a cut-off value) does not remain constant for typical runs. The reasons for this loss or gain of mass are manifold: loss of steepness in the interface region, inexact divergence of the velocity field, boundary velocities, etc. This lack of exact conservation of liquid mass has been reported repeatedly in the literature [5, 11, 49]. The recourse taken here is the classic one: add/remove mass in the interface region in order to obtain an exact conservation of mass. At the end of every timestep, the total amount of fluid mass is compared to the expected value. The expected value is determined from the mass at the previous timestep, plus the mass-flux across all boundaries during the timestep. The differences in expected and actual mass are typically very small (less than 10^{-4}), so that quick convergence is achieved by simply adding and removing mass appropriately. The amount of mass taken/added is made proportional to the absolute value of

the normal velocity of the interface:

$$v_n = \left| \mathbf{v} \cdot \frac{\nabla \Phi}{|\nabla \Phi|} \right| \quad (24)$$

In this way, the regions with no movement of the interface remain unaffected by the changes made to the interface in order to impose strict conservation of mass. The addition and removal of mass typically occurs at points close the liquid–gas interface, where Φ does not assume extreme values. In some instances, the addition or removal of mass can lead to values of Φ outside the allowed range. If this occurs, the value is capped at the extreme value, and further corrections are carried out at the next iteration.

5.5. Deactivation of air region

Given that the air region is not treated/updated, any CPU spent on it may be considered wasted. Most of the work is spent in loops over the edges (upwind solvers, limiters, gradients, etc.). Given that edges have to be grouped in order to avoid memory contention/allow vectorization when forming right-hand sides [50, 51], this opens a natural way of avoiding unnecessary work: form relatively small edge-groups that still allow for efficient vectorization, and deactivate groups instead of individual edges [35]. In this way, the basic loops over edges do not require any changes. The `if`-test whether an edge group is active or deactive occurs outside the inner loops over edges, leaving them unaffected. On scalar processors, edges-groups as small as `negrp` = 8 are used. Furthermore, if points and edges are grouped together in such a way that proximity in memory mirrors spatial proximity, most of the edges in air will not incur any CPU penalty.

5.6. Treatment of bubbles

The treatment of bubbles follows the classic assumption that the timescales associated with speed of sound in the bubble are much faster than the timescales of the surrounding fluid. This implies that at each instance the pressure in the bubble is (spatially) constant. As long as the bubble is not in contact with the atmospheric air (see Figure 3), the pressure can be obtained from the

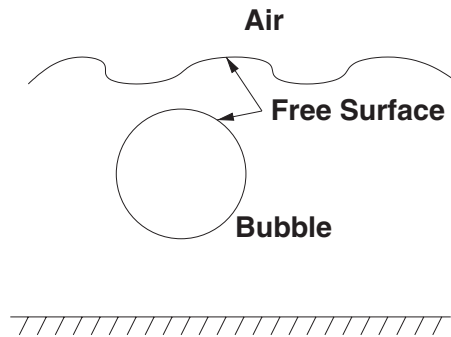


Figure 3. Bubble in water.

isentropic relation

$$\frac{p_b}{p_{b0}} = \left(\frac{\rho_b}{\rho_{b0}} \right)^\gamma \quad (25)$$

where p_b, ρ_b denote the pressure and density in the bubble and p_{b0}, ρ_{b0} the reference values (e.g. those at the beginning of the simulation). The gas in the bubble is marked by solving a scalar advection equation of the form given by Equation (3):

$$b_{,t} + \mathbf{v}_a \cdot \nabla b = 0 \quad (26)$$

where, initially, $b = 1.0$ denotes the bubble region and $b = 0.0$ the remainder of the flowfield. The same advection schemes and steepening algorithms as used for Φ as described above are also used for b . At the beginning of every timestep the total volume occupied by gas is added. From this volume the density is inferred, and the pressure is computed from Equation (25).

At the end of every timestep, a check is performed to see if the bubble has reached contact with the air. This happens if we have, at a given point: $b > 0.5$ and $\Phi > \Phi_{0.5}$. Should this be the case, the neighbour elements of these points that are in air are set to $b = 1.0$. This increases the volume occupied by the bubble, thereby reducing the pressure. Over the course of a few timesteps, the pressure in the bubble then reverts to atmospheric pressure, and one observes a rather quick bubble collapse.

5.7. Adaptive refinement

Adaptive mesh refinement is very often used to reduce CPU and memory requirements without compromising the accuracy of the numerical solution. For transient problems with moving discontinuities, adaptive mesh refinement has been shown to be an essential ingredient of production codes [52, 53]. For multiphase problems the mesh can be refined automatically close to the liquid–gas interface. This has been done in the present case by including two additional refinement indicators (on top of the usual ones based on the flow variables). The first one looks at the edges cut by the liquid–gas interface value of Φ , and refines the mesh to a certain element size or refinement level [54]. The second, more sophisticated indicator, looks at the liquid–gas interface curvature, and refines the mesh only in regions where the element size is deemed insufficient.

6. EXAMPLES

6.1. Breaking dam problem

This is a classic test case for free surface flows. The problem definition is shown in Figure 4(a). This case was run on a coarse mesh with $\text{nelem} = 16\,562$ elements, a fine mesh with $\text{nelem} = 135\,869$ and an adaptively refined mesh (where the coarse mesh was the base mesh) with approximately $\text{nelem} = 30\,000$ elements. The refinement indicator for the latter was the free surface (see above), and the mesh was adapted every 5 timesteps.

Figure 4(b) shows the discretization for the coarse mesh, and Figures 4(c–f) the development of the flowfield and the free surface until the column of water hits the right wall. Note the mesh

SIMULATION OF FLOWS WITH VIOLENT FREE SURFACE MOTION

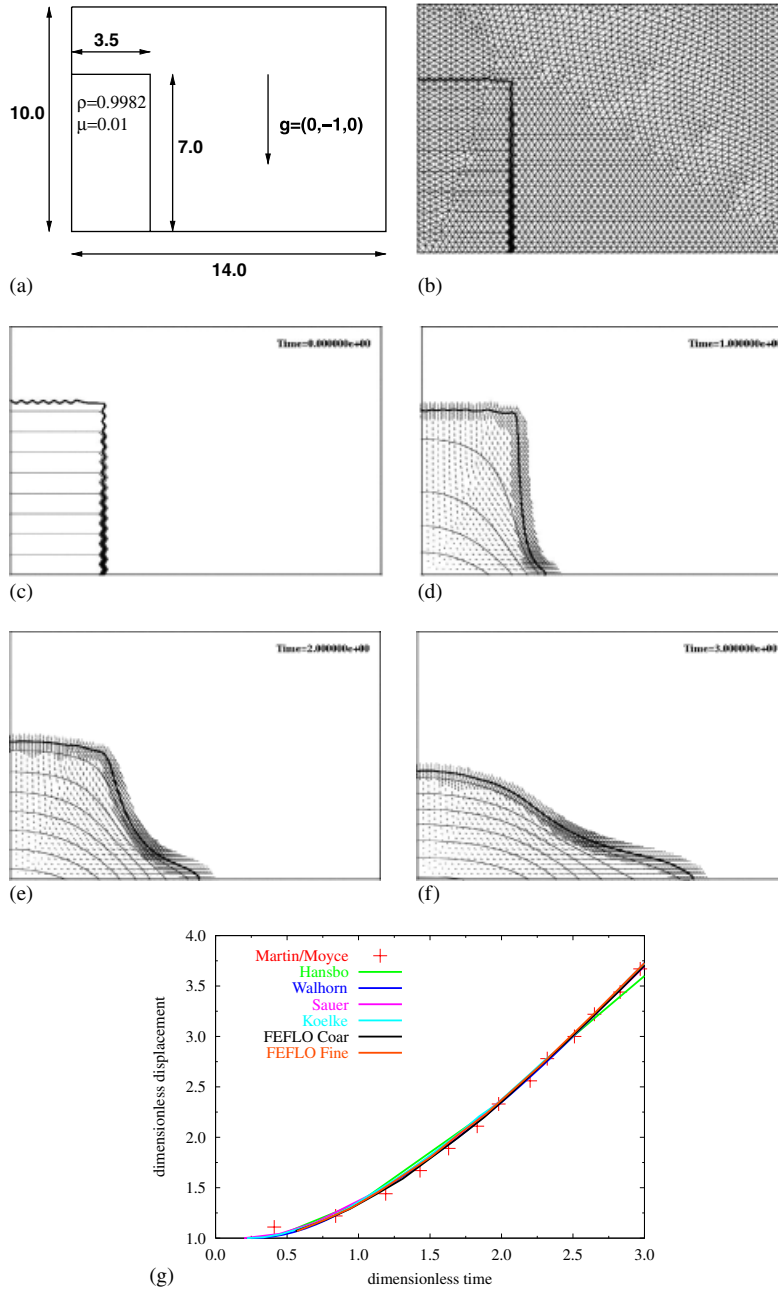


Figure 4. (a) Breaking dam: problem definition; (b) breaking dam: surface discretization for the coarse mesh; (c–f) breaking dam: flowfield at different times; and (g) breaking dam: horizontal displacement.

adaptation in time. The results obtained for the horizontal location of the free surface along the bottom wall are compared to the experimental values of Martin and Moyse [55], as well as the numerical results obtained by Hansbo [56], Kölke [57] and Walhorn [58] in Figure 4(g). The dimensionless time and displacement are given by $\tau = t\sqrt{2g/a}$ and $\delta = x/a$, where a is the initial width of the water column. As one can see, the agreement is very good, even for the coarse mesh. The difference between the adaptively refined mesh and the fine mesh was almost indistinguishable, and therefore only the results for the fine mesh are shown in the graph.

6.2. Sloshing of a 2D tank due to sway excitation

This example considers the sloshing of a partially filled 2D tank.

The main tank dimensions are $L = H = 1$ m, with tank width $B = 0.1$ m. The problem definition is shown in Figure 5(a). Experimental data for this tank with a filling level $h/L = 0.35$ have been provided by Olsen [59], and reported by Faltisen [60] and Olsen and Johnsen [61], where the tank was undergoing a sway motion, i.e. the tank oscillates horizontally with law $x = A \sin(2\pi t/T)$. A wave gage was placed 0.05 m from the right wall and the maximum wave elevation relative to a tank-fixed coordinate system was recorded. In the numerical simulations reported by Landrini *et al.* [62] using the SPH method, the forced oscillation amplitude increases smoothly in time and reaches its steady regime value in 10 T. The simulation continues for another 30 T and the maximum wave elevation is recorded in last 10 periods of oscillation.

We followed the same procedure as Landrini *et al.* [62] in our numerical simulation for 32 cases, which correspond to 2 amplitudes ($A = 0.025, 0.05$) and 16 periods, ranging from $T = 1.0$ – 1.8 s or $T/T_1 = 0.787$ – 1.42 , where $T_1 = 1.27$ s. When $h/L = 0.35$ the primary resonances of the first and the third modes occur at $T/T_1 = 1.0$ and 0.55 , respectively. The secondary resonance of the second mode is at $T/T_1 = 1.28$ (see Reference [62]). The present VOF results for the time history of the lateral force F_x when $T = 1.2, 1.3$ and $A = 0.025, 0.05$ are shown in Figure 5(b). The corresponding time history of the wave elevation at the wave probe A1 (see Figure 5(a)) are shown in Figure 5(c). Some free surface snapshots are shown in Figure 5(d). The dark line represents the free surface. Note also the ‘undershoots’ in the pressure due to extrapolation. The present VOF results for maximum wave elevation ζ at the wave probe A1 (see Figure 5(a)) are compared with the experimental data and SPH results [62] in Figure 5(e) for $A/L = 0.025, 0.05$. We remark that as the wave inclination close to the wall is considerable, there is a non-negligible uncertainty in both the experiments and computational results.

The predicted lateral absolute values of maximum forces are compared with the experimental data and SPH results [62] in Figure 5(f) for $A/L = 0.05$ (there is no force data available for $A/L = 0.025$). Figure 5(g) shows the comparison of predicted lateral absolute values of maximum forces for $A/L = 0.025, 0.05$. It can be seen from Figures 5(e–g) that both maximum wave height and lateral absolute values of maximum forces predicted by present VOF method agrees fairly well with the experimental data and SPH results, with a small phase shift among the three results. Figures 5(b) and (c) are typical time history plots. It should be noted from these figures that even after a long simulation time (40 periods), steady state results are not generally obtained. This is due to very small damping in the system. Landrini *et al.* [62] noted the same behaviour in their numerical simulations. As a result, the predicted maximum

SIMULATION OF FLOWS WITH VIOLENT FREE SURFACE MOTION

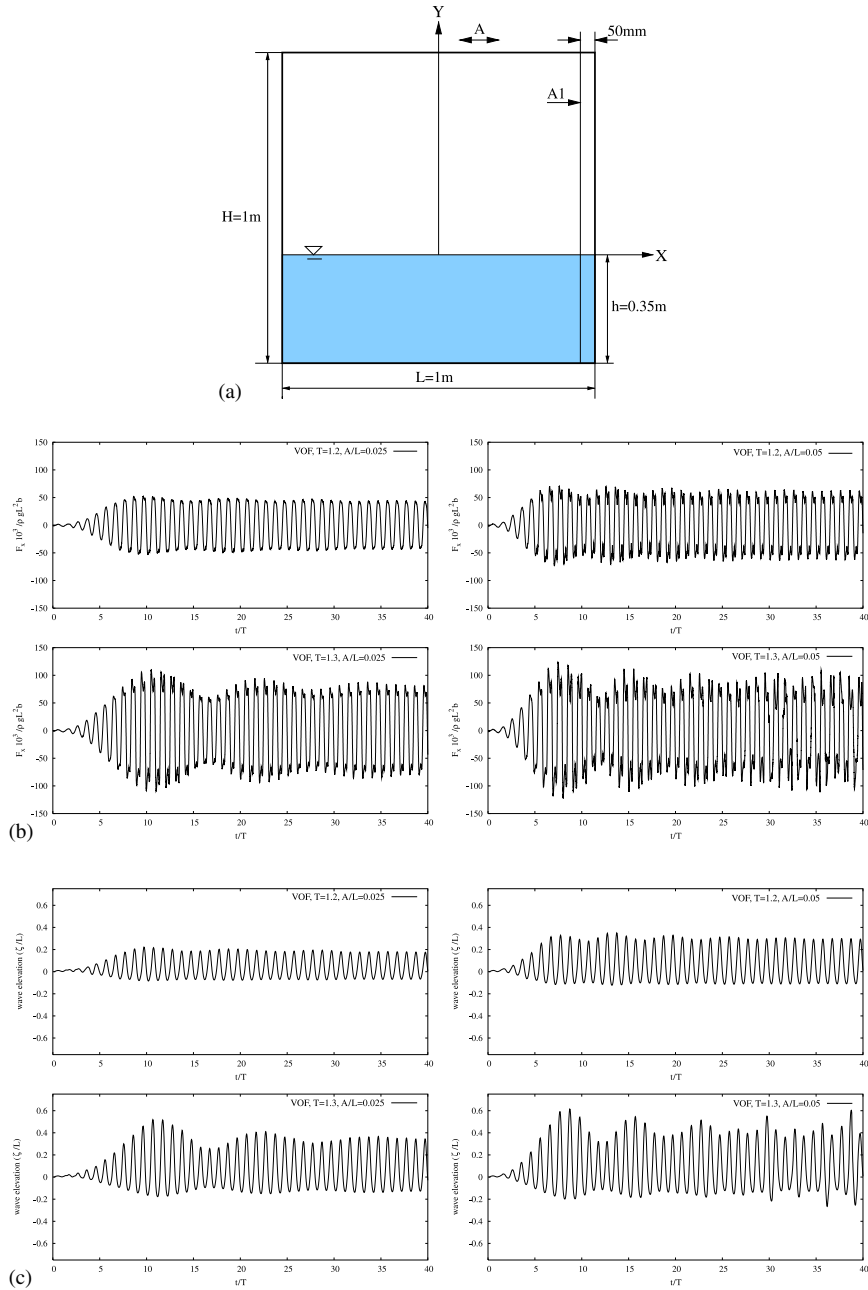


Figure 5. (a) 2D tank: problem definition; (b) 2D tank: time history of lateral force F_x ; (c) 2D tank: time history of wave elevation (probe A1); (d) snapshots of free surface wave elevation for $T=1.3$ and $A/L=0.05$; (e) 2D tank: maximum wave height (probe A1); and (f,g) 2D tank: maximum absolute values of lateral force F_x for $A/L=0.025, 0.05$.

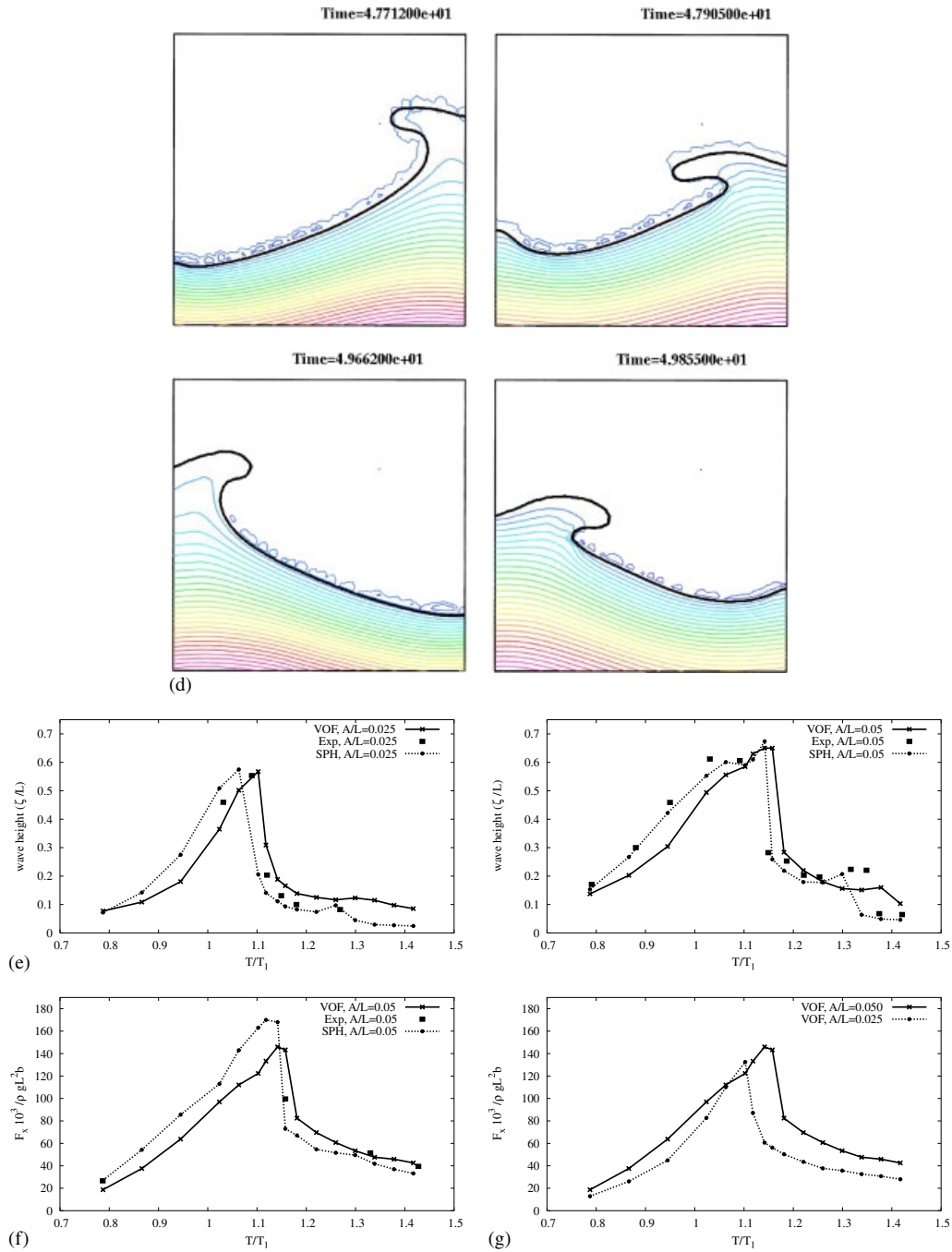


Figure 5. *Continued.*

wave elevation and the lateral absolute values of maximum forces plotted in Figure 5(e) are average maximum values for the last few periods for the cases when the steady state is not reached.

6.3. Sloshing of a 3D tank due to sway excitation

In order to study the 3D effects, the sloshing of a partially filled 3D tank is considered. The main tank dimensions are $L = H = 1$ m, with tank width $b = 1$ m. The problem definition is shown in Figure 6(a). The 3D tank has the same filling level $h/L = 0.35$ as the 2D tank. The 3D tank case is run on a mesh with $n_{elem} = 561\,808$ elements, and the 2D tank is run on a mesh with $n_{elem} = 54\,124$ elements. The numerical simulations are carried out for both 3D and 2D tanks, where both tanks are undergoing the same prescribed sway motion given by $x = A \sin(2\pi t/T)$. The simulations were carried out for $A = 0.025$ and $T = 1.27$ (i.e. $T/T_1 = 1$). The forced oscillation amplitude increases smoothly in time and reaches its steady regime value in 10 T. The simulation continues for another 70 T. In order to show the 3D effects, the forces are non-dimensionalized with $\rho g L^2 b$ for both 2D and 3D tanks. Figures 6(b) and (c) show the time history of the force F_x (horizontal force in the same direction as the tank moving direction) for both 2D and 3D tanks. Figure 6(d) shows the time history of the force F_z (horizontal force perpendicular to the tank moving direction) for 3D tank. It is very interesting to observe from Figures 6(c) and (d) that there are almost no 3D effects for the first 25 oscillating periods. The 3D modes start to appear after 25 T, and fully build up at about 40 T. The 3D flow pattern then remains steady and periodic for the rest of the simulation, which is about 40 more oscillation periods.

Figures 6(e)–(g) show a sequence of snapshots of the free surface wave elevation for the 3D tank. For the first set of snapshots (see Figure 6(e)), the flow is still 2D. The 3D flow starts to build up in the second set of snapshots (see Figure 6(f)). The flow remains periodic 3D for the last 40 periods. Figure 6(g) shows the typical snapshots of the free surface for the last 40 periods. The 3D effects are clearly shown in these plots.

6.4. Drifting ship

This example shows the use of the present methodology to predict the effects of drift in waves for large ships. The problem definition is given in Figure 7(a). The ship is a generic LNG tanker, and is considered rigid. The waves are generated by moving the left wall of the domain. A large element size was specified at the far end of the domain in order to dampen the waves. The mesh at the ‘wave-maker plane’ is moved using a sinusoidal excitation. The ship is treated as a free, floating object subject to the hydrodynamic forces of the water. The surface nodes of the ship move according to a 6 DOF integration of the rigid body motion equations. Approximately 30 layers of elements close to the ‘wave-maker plane’ and the ship are moved, and the Navier–Stokes/VOF equations are integrated using the arbitrary Lagrangian–Eulerian frame of reference. The LNG tanks are assumed 80% full. This leads to an interesting interaction of the sloshing inside the tanks and the drifting ship. The mesh had approximately $n_{elem} = 2\,670\,000$ elements, and the integration to 3 min of real time took 20 h on a PC (3.2 GHz Intel P4, 2 GB RAM, Linux OS, Intel compiler). Figure 7(b) shows the evolution of the flowfield, and Figures 7(c) and (d) the body motion. Note the change in position for the ship, as well as the roll.

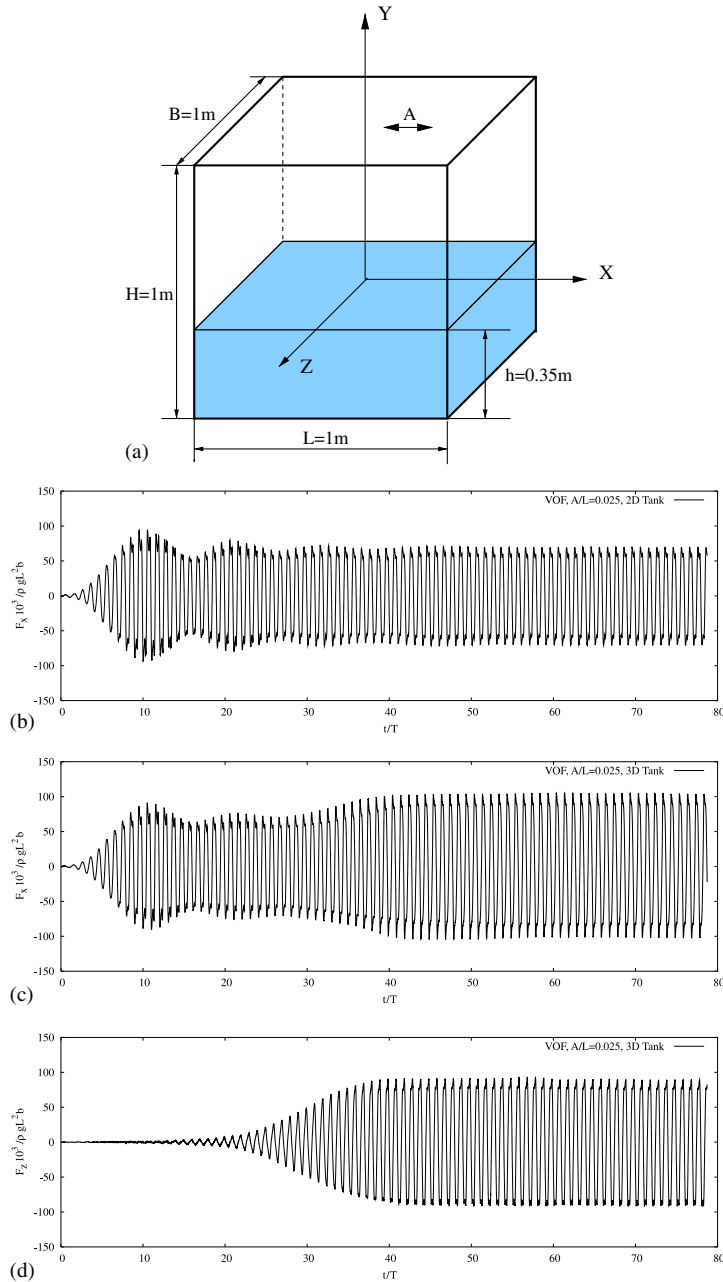


Figure 6. (a) 3D tank: problem definition; (b) 3D tank: time history of force F_x for a 2D tank at $A/L = 0.025$, $T/T1 = 1$; (c) 3D tank: time history of force F_x for a 3D tank at $A/L = 0.025$, $T/T1 = 1$; (d) 3D tank: time history of force F_z for a 3D tank at $A/L = 0.025$, $T/T1 = 1$; (e) snap shots of the free surface wave elevation for 3D tank; (f) snap shots of the free surface wave elevation for 3D tank; and (g) snap shots of the free surface wave elevation for 3D tank.

SIMULATION OF FLOWS WITH VIOLENT FREE SURFACE MOTION

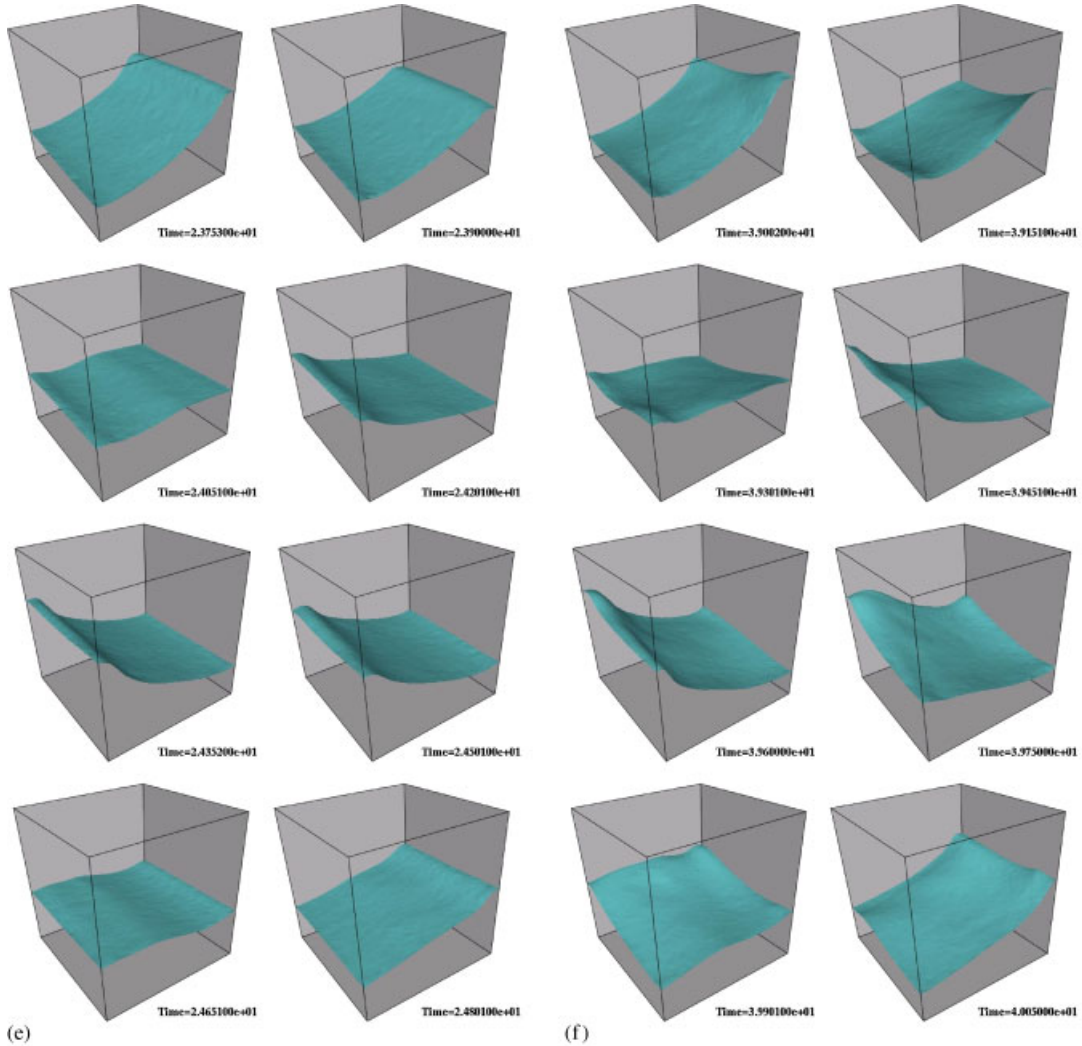


Figure 6. *Continued.*

6.5. Bubble collapse beneath generic ship

This example shows the use of the present methodology to predict the effects of bubble collapse close to structures. The problem definition is given in Figure 8(a). The ship is a generic ferry.

The reference values for the bubble, which was located at midship and approximately 4 m from the hull, were set as follows: volume $V_0 = 128 \text{ m}^3$, density $\rho_0 = 1.25 \text{ kg/m}^3$, pressure $p_0 = 1.0e + 8 \text{ N/m}^2$, polytropic coefficient: $\gamma = 1.4$. The initial radius for the bubble was set to $r = 2 \text{ m}$. The mesh, whose surface is shown in Figure 8(b), had approximately $n_{elem} = 1\,530\,000$ elements. Figures 8(c–g) show the evolution of the flowfield. Note the change of shape for the bubble, first

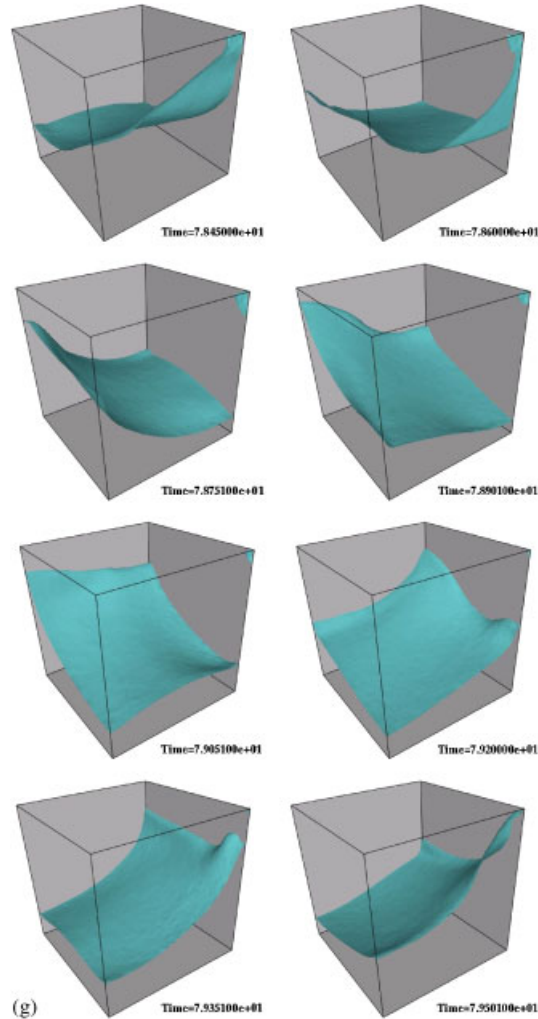


Figure 6. *Continued.*

into a torus and subsequently into a rather complex shape. The pressure recorded at midship on the hull is shown in Figure 8(h).

7. CONCLUSIONS AND OUTLOOK

A volume of fluid (VOF) technique has been developed and coupled with an incompressible Euler/Navier–Stokes solver operating on adaptive, unstructured grids to simulate the interactions of extreme waves and 3D structures. The present implementation follows the classic VOF implementation for the liquid–gas system, considering only the liquid phase. The velocities and pressure in the gas region near the free surface are obtained via extrapolation algorithms. The VOF

SIMULATION OF FLOWS WITH VIOLENT FREE SURFACE MOTION

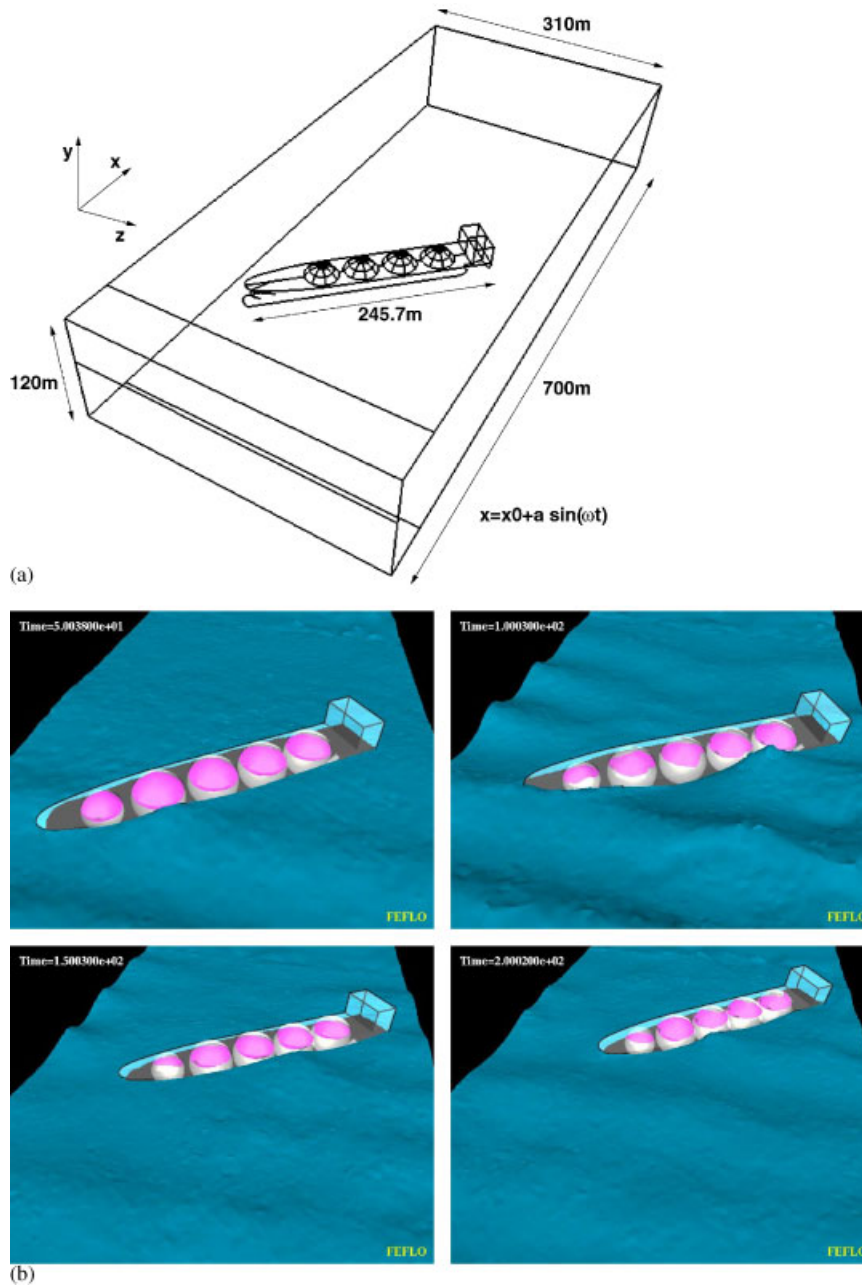
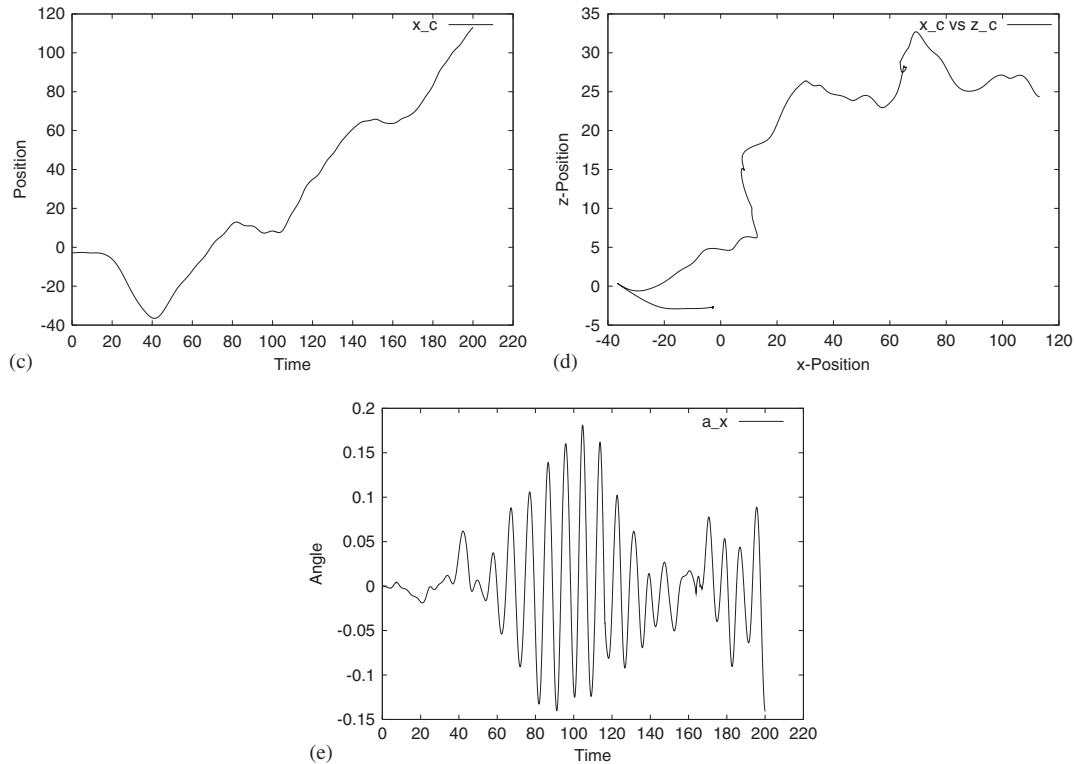


Figure 7. (a) Ship adrift: problem definition; (b) evolution of the free surface; (c–d) position of centre of mass; and (e) roll angle vs time.

Figure 7. *Continued.*

technique was validated against the classic dam-break problem, as well as series of 2D sloshing experiments and results from SPH calculations. Other examples presented include a ship drifting in waves and a bubble collapsing under a generic ship.

When taken together, these recent advances, which include:

- accurate, fast incompressible Navier–Stokes solvers operating on adaptive, unstructured grids;
- robust volume of fluid (VOF) techniques for free surface flows;
- deactivation techniques to speed up calculations; and
- extensive parallelization of solvers

have made it possible to simulate flows with violent free surface motion with a high degree of accuracy, allowing decision-making based on them. Like every human endeavour, numerical algorithms are subject to continuous improvements. Present research is directed at the proper treatment of

- surface tension;
- incoming and outgoing waves for 3D VOF-based free surface flows;
- free surface wall boundary conditions for RANS, NS cases (i.e. those cases where the velocity at the wall $\mathbf{v} = 0$); and
- multiple bubble interaction (splitting, merging, etc.).

SIMULATION OF FLOWS WITH VIOLENT FREE SURFACE MOTION

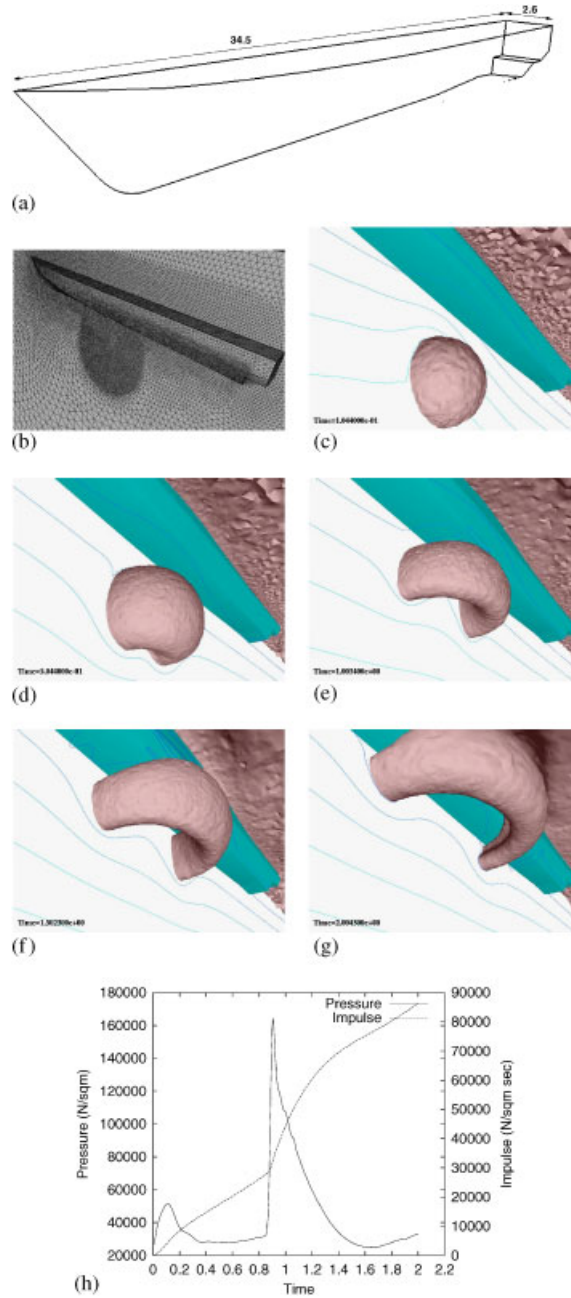


Figure 8. (a) Bubble collapse: problem definition; (b–g) surface mesh and evolution of bubble; and (h) pressure and impulse recorded at midship on hull.

ACKNOWLEDGEMENTS

A considerable part of this work was carried out at the International Center for Numerical Methods in Engineering (CIMNE) of the Universidad Polit cnica de Catalunya, Barcelona, Spain. The support for this visit is gratefully acknowledged.

The authors also wish to thank Mr Andrea Colagrossi of the INSEAN for providing both experimental data and numerical results for the sloshing of a 2D tank.

REFERENCES

1. Nichols BD, Hirt CW. Methods for calculating multi-dimensional, transient free surface flows past bodies. *Proceedings of the First International Conference on Numerical Ship Hydrodynamics*, Gaithersburg, ML, 20–23 October 1975.
2. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**:201–225.
3. Yabe T, Aoki T. A universal solver for hyperbolic equations by cubic-polynomial interpolation. *Computer Physics Communications* 1991; **66**:219–242.
4. Unverdi SO, Tryggvason G. A front tracking method for viscous incompressible flows. *Journal of Computational Physics* 1992; **100**:25–37.
5. Sussman M, Smereka P, Osher S. A levelset approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**:146–159.
6. Yabe T. Universal solver CIP for solid, liquid and gas, Chapter 3. *Computational Fluid Dynamics Review*, Hafez MM, Oshima K (eds). Wiley, 1997.
7. Scardovelli R, Zaleski S. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics* 1999; **31**:567–603.
8. Chen G, Kharif C. Two-dimensional Navier–Stokes simulation of breaking waves. *Physics of Fluids* 1999; **11**(1):121–133.
9. Fekken G, Veldman AEP, Buchner B. Simulation of green water loading using the Navier–Stokes equations. *Proceedings of the 7th International Conference on Numerical Ship Hydrodynamics*, Nantes, France, 1999.
10. B  usser B, Fraunie P, Grilli S, Marcer R. Numerical analysis of the internal kinematics and dynamics of three-dimensional breaking waves on slopes. *International Journal of Offshore and Polar Engineering* 2004; **14**(4):247–256.
11. Enright D, Nguyen D, Gibou F, Fedkiw R. Using the particle level set method and a second order accurate pressure boundary condition for free surface flows. *Proceedings of the 4th ASME–JSME Joint Fluids Engineering Conference*, FEDSM2003-45144, Kawahashi M, Ogut A, Tsuji Y (eds). Honolulu, HI, 2003; 1–6.
12. Huijsmans RHM, van Groen E. Coupling freak wave effects with green water simulations. *Proceedings of the 14th ISOPE*, Toulon, France, 23–28 May 2004.
13. Coppola-Owen AH, Codina R. Improving Eulerian two-phase flow finite element approximation with discontinuous gradient pressure shape functions. *International Journal for Numerical Methods in Fluids* 2005.
14. Codina R, Soto O. A numerical model to track two-fluid interfaces based on a stabilized finite element method and the level set technique. *International Journal for Numerical Methods in Fluids* 2002; **4**:293–301.
15. L  hner R. A fast finite element solver for incompressible flows. *AIAA-90-0398*, 1990.
16. Martin D, L  hner R. An implicit linelet-based solver for incompressible flows. *AIAA-92-0668*, 1992.
17. Ramamurti R, L  hner R. A parallel implicit incompressible flow solver using unstructured meshes. *Computers and Fluids* 1996; **5**:119–132.
18. L  hner R, Yang C, O  ate E, Idelsohn S. An unstructured grid-based, parallel free surface solver. *Applied Numerical Mathematics* 1999; **31**:271–293.
19. L  hner R. Multistage explicit advective prediction for projection-type incompressible flow solvers. *Journal of Computational Physics* 2004; **195**:143–152.
20. Kim J, Moin P. Application of a fractional-step method to incompressible Navier–Stokes equations. *Journal of Computational Physics* 1985; **59**:308–323.
21. Bell JB, Colella P, Glaz H. A second-order projection method for the Navier–Stokes equations. *Journal of Computational Physics* 1989; **85**:257–283.
22. Bell JB, Marcus DL. A second-order projection method for variable-density flows. *Journal of Computational Physics* 1992; **101**(2):334–348.

23. Alessandrini B, Delhommeau G. A multigrid velocity-pressure-free surface elevation fully coupled solver for calculation of turbulent incompressible flow around a hull. *Proceedings of the 21st Symposium on Naval Hydrodynamics*, Trondheim, Norway, June 1996.
24. Kallinderis Y, Chen A. An incompressible 3-D Navier–Stokes method with adaptive hybrid grids. *AIAA-96-0293*, 1996.
25. Gresho PM, Upson CD, Chan ST, Lee RL. Recent progress in the solution of the time-dependent, three-dimensional, incompressible Navier–Stokes equations. *Proceedings of the 4th International Symposium on Finite Element Methods in Flow Problems*, Kawai T (ed.). University of Tokyo Press, 1982; 153–162.
26. Donea J, Giuliani S, Laval H, Quartapelle L. Solution of the unsteady Navier–Stokes equations by a fractional step method. *Computer Methods in Applied Mechanics and Engineering* 1982; **30**:53–73.
27. Gresho PM, Chan ST. On the theory of semi-implicit projection methods for viscous incompressible flows and its implementation via a finite element method that introduces a nearly—consistent mass matrix. *International Journal for Numerical Methods in Fluids* 1990; **11**:621–659.
28. Takamura A, Zhu M, Vinteler D. Numerical simulation of pass-by maneuver using ALE technique. *JSAE Annual Conference*, Spring, Tokyo, May 2001.
29. Eaton E. Aero-acoustics in an automotive HVAC module. *American PAM User Conference*, Birmingham, Michigan, 24–25 October 2001.
30. Karbon KJ, Kumarasamy S. Computational aeroacoustics in automotive design, computational fluid and solid mechanics. *Proceedings of the First MIT Conference on Computational Fluid and Solid Mechanics*, Boston, June 2001; 871–875.
31. Codina R. Pressure stability in fractional step finite element methods for incompressible flows. *Journal of Computational Physics* 2002; **170**:112–140.
32. Li Y, Kamioka T, Nouzawa T, Nakamura T, Okada Y, Ichikawa N. Verification of aerodynamic noise simulation by modifying automobile front-pillar shape. *JSAE 20025351*, *JSAE Annual Conference*, Tokyo, July 2002.
33. Karbon KJ, Singh R. Simulation and design of automobile sunroof buffeting noise control. *8th AIAA-CEAS Aero-Acoustics Conference*, Breckridge, June 2002.
34. Camelli F, Löhner R, Sandberg WC, Ramamurti R. VLES study of ship stack gas dynamics. *AIAA-04-0072*, 2004.
35. Löhner R. *Applied CFD Techniques*. Wiley: New York, 2001.
36. Hufnagel JD, Khaletsky D. A finite element method to solve the Navier–Stokes equations using the method of characteristics. *International Journal for Numerical Methods in Fluids* 1984; **4**:247–269.
37. Gregoire JP, Benque JP, Lasbleiz P, Goussebaile J. 3-D industrial flow calculations by finite element method. *Springer Lecture Notes in Physics* 1985; **218**:245–249.
38. Kelly DW, Nakazawa S, Zienkiewicz OC, Heinrich JC. A note on anisotropic balancing dissipation in finite element approximation to convection diffusion problems. *International Journal for Numerical Methods in Engineering* 1980; **15**:1705–1711.
39. Brooks AN, Hughes TJR. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
40. Gunzburger MD. Mathematical aspects of finite element methods for incompressible viscous flows. In *Finite Elements: Theory and Application*, Dwyer, Hussaini, Voigt (eds). Springer: Berlin, 1987; 124–150.
41. Fortin M, Thomasset F. Mixed finite element methods for incompressible flow problems. *Journal of Computational Physics* 1979; **31**:113–145.
42. Soulaimani A, Fortin M, Ouellet Y, Dhatt G, Bertrand F. Simple continuous pressure elements for two- and three-dimensional incompressible flows. *Computer Methods in Applied Mechanics and Engineering* 1987; **62**:47–69.
43. Thomasset F. *Implementation of Finite Element Methods for Navier–Stokes Equations*. Springer: Berlin, 1981.
44. Taylor C, Hood P. A numerical solution of the Navier–Stokes equations using the finite element method. *Computational Fluids* 1973; **1**:73–100.
45. Franca LP, Hughes TJR, Loula AFD, Miranda I. A new family of stable elements for the Stokes problem based on a mixed Galerkin/least-squares finite element formulation. In *Proceedings of the 7th International Conference on Finite Elements in Flow Problems*, Chung TJ, Karr F (eds). Huntsville, AL, 1989; 1067–1074.
46. Tezduyar TE, Shih R, Mittal S, Ray SE. Incompressible flow computations with stabilized bilinear and linear equal-order interpolation velocity-pressure elements. *UMSI Report 90*, 1990.
47. Franca LP, Frey SL. Stabilized finite element methods: II. The incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1992; **99**:209–233.

48. Sweby PK. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis* 1984; **21**:995–1011.
49. Sussman M, Puckett E. A coupled level set and volume of fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics* 2000; **162**:301–337.
50. Löhner R. Some useful renumbering strategies for unstructured grids. *International Journal for Numerical Methods in Engineering* 1993; **36**:3259–3270.
51. Löhner R. Renumbering strategies for unstructured-grid solvers operating on shared-memory, Cache-based parallel machines. *Computer Methods in Applied Mechanics and Engineering* 1998; **163**:95–109.
52. Baum JD, Luo H, Mestreau E, Löhner R, Pelessone D, Charman C. A coupled CFD/CSD methodology for modeling weapon detonation and fragmentation. *AIAA-99-0794*, 1999.
53. Löhner R, Yang C, Baum JD, Luo H, Pelessone D, Charman C. The numerical simulation of strongly unsteady flows with hundreds of moving bodies. *International Journal for Numerical Methods in Fluids* 1999; **31**:113–120.
54. Löhner R, Baum JD. Adaptive H-refinement on 3-D unstructured grids for transient problems. *International Journal for Numerical Methods in Fluids* 1992; **14**:1407–1419.
55. Martin JC, Moyce WJ. An experimental study of the collapse of a liquid column on a rigid horizontal plane. *Philosophical Transactions on Royal Society of London, Series A* 1952; **244**:312–324.
56. Hansbo P. The characteristic streamline diffusion method for the time-dependent incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1992; **99**:171–186.
57. Kölke A. Modellierung und Diskretisierung bewegter Diskontinuitäten in Randgekoppelten Mehrfeldaufgaben. *Ph.D. Thesis*, Braunschweig, TU, 2005.
58. Walhorn E. Ein Simultanes Berechnungsverfahren für Fluid-Struktur-Wechselwirkungen mit Finiten Raum-Zeit-Elementen. *Ph.D. Thesis*, Braunschweig, TU, 2002.
59. Olsen H. Unpublished Sloshing Experiments at the Technical University of Delft. Delft, The Netherlands, 1970.
60. Faltisen OM. A nonlinear theory of sloshing in rectangular tanks. *Journal of Ship Research* 1974; **18/4**:224–241.
61. Olsen H, Johnsen KR. Nonlinear sloshing in rectangular tanks. A pilot study on the applicability of analytical models. *Det Norske Veritas Report 74-72-S*, vol. II, 1975.
62. Landrini M, Colagorossi A, Faltisen OM. Sloshing in 2-D flows by the SPH method. *Proceedings of the 8th International Conference on Numerical Ship Hydrodynamics*, Busan, Korea, 2003.